Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 Программирование в компьютерных системах

**ОТЧЕТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**(по профилю специальности)**

Профессиональный модульПМ.02 Разработка и администрирование баз данных

*(наименование профессионального модуля)*

Выполнил:

обучающийся учебной группы № 120

Т.Р. Добряков

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

И.В. Сибирев

*(И.О. Фамилия)*

**Москва**

**20­­­­23**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc122556346)

[ОБЩАЯ ЧАСТЬ 5](#_Toc122556347)

[Глава 1 5](#_Toc122556348)

[Глава 2 6](#_Toc122556349)

[ЗАКЛЮЧЕНИЕ 13](#_Toc122556350)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc122556351)

[ПРИЛОЖЕНИЕ 15](#_Toc122556352)

# ВВЕДЕНИЕ

Компания «Интеграл Компьютерных Технологий» основана в 2002 году.

Компания входит в группу компаний MBG

В составе команды более 150 сертифицированных экспертов и IT-специалистов с опытом работы от 10-25 лет в различных отраслях экономики.

Компания имеет статус 1С:Франчайзинг.

Обеспечивает оперативную IT-поддержку по всей России для многих бизнес-решений в режиме 24/7/365.

СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА

распространяется на предоставление комплексных услуг по автоматизации производственных и офисных процессов, учета и отчётности на предприятии заказчика, включая: консультационные услуги, помощь в выборе и продажу программных продуктов, проектирование, разработку, внедрение и сопровождение информационных систем, индивидуальное и групповое обучение пользователей.

Ведущие программисты 1С Предприятие 8.3 компании имеют:

* Опыт анализа, разработки, проектирования и сопровождения автоматизированных информационных систем на платформе 1С 8.3.
* Опыт в разработке и оптимизации различных обменов.
* Умение встраиваться в код типовых конфигураций.
* Опыт разработки для высоконагруженных систем;
* Опыт работы с большими БД MS SQL;
* Опыт работы с управляемыми формами;

Компания занимается внедрением и поддержкой систем, интеграцией, консалтингом и обучением:

1. Типовые:

* 1С: Корпорация
* 1С: ERP Управление предприятием 2
* 1С: ERP. Управление холдингом
* 1С: Управление торговлей
* 1С: Бухгалтерия
* 1С: Розница
* 1С: Зарплата и управление персоналом ред. 3.1
* 1С: Документооборот КОРП

1. Отраслевые:

* 1С: Медицинская информационная система. Поликлиника
* 1С: Медицина. Больница
* 1С: MDM (Master Data Management-НСИ)
* 1С: Медицина. Больничная аптека 2.0
* 1С: Фитнес-клуб КОРП

Диаграмма информационных потоков [см. Приложение 10]

Диаграмма бизнес процессов [см. Приложение 11]

# ОБЩАЯ ЧАСТЬ

## Глава 1

Для прохождения производственной практики мне был выдан методический материал для изучения .

Поставлена задача изучить и закрепить на практике следующие темы:

* Основная информация и термины 1С.
* Изучение архитектуры 1С.
* Изучение основных функций 1С:Предприятние.
* Изучение процедур и функций 1С:Предприятие.
* Изучение и формирование отчётов в 1С:Предприятие.
* Стандарты разработки и проектирования на платформе 1С.

## Глава 2

Основная информация и термины 1С

* 1С:Предприятие — программный продукт компании «1С», предназначенный для автоматизации деятельности на предприятии.
* "Платформа" - это продукт компании. Это ещё не конкретная программа для пользователя, но совокупность всех инструментов, которые нужны для её работы. Здесь есть IDE для разработчика, приложения для администрирования, клиентская часть (интерпретатор). Всё это в отдельности имеет свои названия, но чаще всего можно просто обощённо сказать "Платформа" и все 1Сники тебя поймут.
* "Конфигурация" - это конкретный программный продукт. Это, грубо говоря, проект в Visual Studio. При этом под конфигурацией могут понимать и ещё конкретное написанное приложение от 1С. Т.е. тиражный проект, который поставляется с открытым исходным кодом. Например, Управление Торговлей или Бухгалтерия предприятия. Пользователь получает специальный шаблон установки, который разворачивает из типового проекта ваше приложение.
* 1С - Это программная оболочка над базой данных. По сути, приложение 1С невозможно без базы данных по той причине, что сам проект (конфигурация) хранится в той же БД, где и лежат данные . Но об этом подробнее поговорим далее.
* Кроссплатформа. Одно и то же решение можно запустить на Windows, Mac, Linux. Можно запустить вообще без установки платформы - в браузере (причем и с телефона). И в большинстве случаев код будет один и тот же (за исключением некоторых нюансов). Так же есть возможность написания специального приложения под Android и iOS. Это может быть и приложение-клиент (когда пользователь работает с той же базой напрямую) или же отдельное приложение, которое синхронизируется с основной базой.
* Интеграция с внешними базами, веб-сервисами и мессенджерами. Это и подключение к другой БД "напрямую" (когда пользователи работают с ней не покидая 1С), и публикация WebAPI для работы из сторонних приложений, и разработка ботов для телеграм.
* Гибкий пользовательский интерфейс. Без необходимости дополнительной разработки пользователь может полностью перенастроить интерфейс под свои нужды. При этом всё сделано ненавязчиво. Если пользователю такое не нужно, то он просто не будет открывать соответствующие пункты меню (часто про них мало кто вообще знает). Но если понадобится, то волен сам настроить: отборы, сортировки, группировки, оформление, отображаемые поля и так далее. Естественно, разработчик может что-то запретить к изменению, но чаще всего это не требуется.

Изучение архитектуры платформы 1С

* Основная задача платформы заключается в повышении уровня абстракции при разработке и использовании прикладных решений. Это позволяет перейти от технических и низкоуровневых понятий к более содержательным и высокоуровневым. Позволяет приблизить эти понятия к языку пользователей и специалистов в предметной области. В конечном итоге это значительно ускоряет и унифицирует разработку прикладного решения и его сопровождение.

Одновременно с этим платформа решает и традиционные задачи, связанные с производительностью, эргономикой, функциональностью и т. д.

* Платформа и прикладные решения

В системе «1С:Предприятие 8» существует четкое разделение на платформу и прикладное решение. Платформа представляет собой framework, в котором функционирует прикладное решение:

Платформа служит фундаментом для построения прикладных решений,

Платформа является средой их исполнения,

Платформа содержит инструментарий, необходимый для разработки, администрирования и поддержки прикладных решений.

При этом прикладное решение является самостоятельной сущностью и может выступать в качестве отдельного программного продукта. Но полностью опирается на технологии платформы.

* Метаданные — способ описания прикладного решения

Прикладное решение не пишется в прямом смысле на языке программирования. Язык программирования используется только там, где это действительно необходимо.

В основе прикладного решения лежат метаданные. Они представляют собой структурированное декларативное его описание. Метаданные образуют иерархию объектов, из которых формируются все составные части прикладной системы и которые определяют все аспекты ее поведения. Фактически, при работе прикладного решения, платформа «проигрывает» (интерпретирует) метаданные, обеспечивая всю необходимую функциональность.

Метаданными описываются структуры данных, состав типов, связи между объектами, особенности их поведения и визуального представления, система разграничения прав доступа, пользовательский интерфейс и т. д. В метаданных сосредоточены сведения не только о том, «что хранить в базе данных», но и о том, «зачем» хранится та или иная информация, какова ее роль в системе, и как связаны между собой информационные массивы.

Использование языка программирования ограничено решением тех задач, которые действительно требуют алгоритмического описания, например, расчета налогов, проверки корректности введенных данных и т. д.

* Построение прикладного решения на основе модели

В платформе заложена ориентация на построение прикладного решения на основе определенной модели. Под моделью понимается вся идеология построения прикладного решения. Сюда относятся способы построения структур данных, типы связей между данными, принципы манипулирования данными, формы описания бизнес-логики, способы связи данных с интерфейсными объектами, разделение функциональности по уровням системы и многое другое.

Важно, что все прикладные решения следуют принятой модели и этим обеспечивается единообразие и предсказуемость их поведения.

* Стандартные прототипы прикладных объектов

В модели разработки «1С:Предприятия 8» все прикладное решение описывается метаданными в виде совокупности прикладных объектов, выбираемых из определенного набора прототипов (классов). Каждый такой прототип отвечает за отражение в прикладном решении определенной совокупности объектов или процессов предметной области, имеющих схожие поведенческие характеристики и сходную роль в общей картине решения.

Примерами таких прототипов являются «Справочники», «Документы», «Регистры накопления».

Каждый прототип имеет некоторую базовую реализацию, которая определяет особенности функционирования создаваемых на основе данного прототипа объектов: структуру хранимых сущностей вместе с некоторыми предопределенными полями, набор типов языка программирования, методы, свойства и события, а также типовые, для решаемой задачи, операции, способы отображения и редактирования, методы регулирования прав доступа и т. д.

Таким образом, все прикладное решение, фактически, состоит из объектов, четко разделенных по тем ролям, которые они играют в прикладном решении. Такой подход существенно усиливает эффект и от описания системы в терминах метаданных, и от построения приложения на основе модели. Подробнее…

* Согласованность технологий и инструментов

Ключевым качеством платформы является достаточность ее средств для решения задач, стоящих перед прикладными решениями. Это позволяет обеспечить очень хорошую согласованность всех технологий и инструментов, которыми пользуется разработчик.

[См. Приложение 12]

Основные функции 1С:Предприятие

* В зависимости от набора установленных модулей «Предприятие» может выполнять несколько функций:
* автоматизация организационной и хозяйственной деятельности;
* учет и аналитика по организации;
* планирование, ведение бюджета, анализ финансов предприятия;
* бухгалтерский учет, отчетность, управление несколькими счетами;
* расчет зарплаты и других выплат сотрудникам;
* управление персоналом.

Руководитель организации может самостоятельно определить необходимые для его платформы функции в зависимости от целей и групп пользователей, которые будут допущены к системе.

Процедуры и функции

* Функции в 1С выполняют некий код и возвращают некое значение. Процедуры же ничего не возвращают. При этом объявлять типы нет никакой возможности. Костыльным путём в новой IDE пытаются применить практику "раз ЯП не может, то будем парсить комментарии". Но фактически на исполнение кода эти комментарии никак не влияют и нужны только для удобства разработки.
* Здесь простая функция для демонстрации. Принимает на вход два параметра. Первый обязателен и без него вызвать функцию нельзя. Второй же можно пропустить и тогда будет использоваться значение "2". Внутри функции идёт объявление переменной "Результат" с присвоением 0. А далее цикл for от нуля до значения первого параметра. И вызывающий функцию получит значение переменной Результат.

МойРезультат = ИмяФункции(100);

[См.Приложение 13]

* 1С обладает Клиент-Серверной архитектурой. Весь код пишется с учетом того, где именно он будет выполняться. И для этого используются директивы компиляции и предпрооцессора (&НаКлиенте, &НаСервере и так далее). Когда разработчик пишет свою процедуру, то он сразу описывает где она будет выполняться. Может на клиенте, может на сервере, а может и там и тут. Последний вариант имеет ограничения, ведь такой код должен одновременно подходить под оба первых варианта.

Изучение и формирование отчётов

* Отчет - это класс для вывода пользователям сводной информации.

Создадим отчет "ОстаткиТоваров" (названия классов должны быть уникальны только в рамках своего базового класса). И сразу нажмём "Открыть схему компоновки данных"

[См. Приложение 14]

* Схема компоновки данных - это один из ключевых механизмов платформы 1С. Она позволяет быстро разрабатывать отчеты, которые пользователь сможет гибко донастроить под свои нужды. Сначала создадим новый набор данных - Запрос. И откроем конструктор запроса

[См. Приложение 15]

* Язык запросов 1С повторяет часть возможностей языка запросов SQL. Далее запрос интерпретируется платформой в SQL'ный. С одним принципиальным отличием. 1Сный язык только для выборки данных. Изменять данные в БД запросами нельзя. Изменение происходит только с использованием объектной модели.
* Конструктор запросов позволяет интерактивно построить запрос. Слева доступные таблицы в БД. По центру выбранные таблицы. А справа выбранные поля. Ещё много закладок, но об этом когда-нибудь потом.

[См. Приложение 16]

Сохраним такой запрос

Вот так выглядит текст запроса:

[См. Приложение 17]

* Сначала донастроим отчет. Выведем теперь все поля пользователю:

[См. Приложение 18]

Работа с отчетом

* Созданный отчет автоматически наполнился всем нужным пользователю функционалом. Настройки и их сохранение и загрузка, поиск по результату и вывод в файл или на принтер. Нам больше ничего делать не нужно и достаточно просто сформировать отчет.

[См. Приложение 19]

* Вот такой простой результат. Он показывает в разрезе товара сколько осталось. Сделаем ещё один отчет "ВсеДвиженияТоваров". И в качестве запроса укажем просто вывод всего из таблицы регистра:

[См. Приложение 20]

Для наглядности я создал ещё несколько документов.

[См. Приложение 21]

Нажмём "Изменить вариант":

[См. Приложение 22]

* Это пользовательский интерфейс по настройке отчета. В нём можно переопределить полностью выводимые поля, структуру, отборы, сортировку и оформление. Добавить свои поля (используя язык выражений 1С).
* Отключил ненужные мне поля и вынес дополнительно Код из поля Товар, Организацию из поля Регистратор (документ) и ИНН этой организации. Обращение "через точку" встречается в 1С повсеместно и в режиме пользователя.
* Сформируем наш вариант отчета. Вот так пользователь быстро подстроил под себя выводимые данные:

[См. Приложение 23]

Компоновка данных

* Но как так выходит, что мы писали один запрос (с выборкой данных из таблицы регистра), а пользователь в результате выводит ещё и данные из таблицы документов, товаров и организаций? Дело в том, что система компоновки данных в 1С автоматически дорабатывает текст запроса так, чтобы вывести пользователю всё то, что он хочет. И наш запрос превратился в это:

[См. Приложение 24]

* В запросе выбираются те поля, которые нужны. И не выбираются те, которые отключены. А так же выбираются поля "через точку". Товар.Код, Организация.ИНН и так далее.

Дело в том, что хоть язык запросов 1С и сделан по образу и подобию SQL, но при этом позволяет ещё делать "обращения через точку". Откроем запрос в конструкторе:

[См. Приложение 25]

* В конструкторе наглядно видно как поле товар, содержащее ссылку на справочник, можно "развернуть" и вытянуть содержимое. Так работают запросы с любыми ссылочными типами. И такой запрос на уровне SQL будет превращён в соединение разных таблиц.

[См. Приложение 26]

* Обращение "через точку" используется повсеместно. Вот так выглядит возможность пользовательского отбора в отчете. Доступны все поля (включая подчиненные). Множество вариантов сравнения. Группировки отборов и так далее. Всё это платформа делает сама.

Вот такой запрос получается:

[См. Приложение 27]

* В языке запросов 1С специальные конструкции, которые описывают, что может пользовать изменить, а что - нет. Например, вот так можно описать состав разрешенных полей для отборов. При этом Товар.\* позволяет делать отборы на реквизиты товара. А вот обратиться к реквизитам регистратора уже нельзя (можно только указывать конкретные ссылки на документы):

[См. Приложение 28]

Стандарты разработки и проектирования на платформе 1С

Стандарты

Система стандартов и методик разработки конфигураций для платформы 1С:Предприятие 8 является частью Инструкций по разработке на 1С

* Создание и изменение объектов метаданных
* Реализация обработки данных
* Соглашения при написании кода
* Клиент-серверное взаимодействие
* Общие вопросы безопасности
* Настройка прав доступа к данным
* Реализация обмена данными
* Разработка и использование библиотек
* Требования по локализации
* Проектирование интерфейсов для 8.3
* Разработка пользовательских интерфейсов
* Проектирование интерфейсов для 8.2
* Разработка пользовательских интерфейсов (обычное приложение)

Создание и изменение объектов метаданных

Содержит рекомендации и методики по организации работы конфигурации, способах хранения данных и их обработки

Основные разделы:

* Организация работы конфигурации
* Учет версий конфигураций
* Организация хранения данных
* Обработчики событий объектов
* Регламентные задания

Реализация обработки данных

Данный раздел содержит рекомендации и методики по работе с самими данными это касается получения данных различными способами, из изменение и удаление, а так же вопросы оптимизации при доступе к данным.

Основные разделы:

* Работа с запросами
* Оптимизация запросов
* Обработка и модификация данных
* Избыточные блокировки и методы оптимизации

Соглашение при написании кода

Свод правил по работе со встроенным языком рассматривает вопросы оформления кода, именования процедур функций и переменных, правильного использования языковых конструкций и прикладных объектов и универсальных коллекций платформы.

Основные разделы:

* Оформление модулей
* Использование конструкций встроенного языка
* Использование прикладных объектов и универсальных коллекций значений

Клиент-серверное взаимодействие

Методики и рекомендации направленные на укрепление клиент-серверной модели взаимодействия путем уменьшения трафика между клиентом и севером, облегчения работы клиента и загрузки серверных мощностей.

Основные разделы:

* Использование модулей с повторным использованием возвращаемых значений
* Использование значений, влияющих на поведение клиентского приложения
* Получение предопределенных значений на клиенте
* Минимизация количества серверных вызовов и трафика
* Минимизация кода, выполняемого на клиенте
* Доступ к файловой системе из кода конфигурации
* Оптимизация использования оперативной памяти
* Таймауты при работе с внешними ресурсами

Общие вопросы безопасности

Раздел направлен на обеспечение безопасной работы системы, предотвращения деструктивных случайных или намеренных действий пользователя или третих лиц

Основные разделы:

* Безопасность прикладного программного интерфейса сервера
* Ограничение на установку признака «Вызов сервера» у общих модулей
* Безопасное хранение паролей
* Ограничение на выполнение внешнего кода
* Ограничения на использование Выполнить и Вычислить на сервере
* Безопасность запуска приложений
* Безопасность программного обеспечения, вызываемого через открытые интерфейсы

Настройка прав доступа к данным

Описание безопасных и оптимальных подходов к разграничению доступа к данным

Основные разделы:

* Настройка ролей и прав доступа
* Стандартные роли
* Установка прав для новых объектов и полей объектов
* Проверка прав доступа
* Использование привилегированного режима
* Ограничения на использование ключевого слова "РАЗРЕШЕННЫЕ" в запросах
* Влияние изменения значений параметров сеанса и функциональных опций на производительность механизма ограничения доступа к данным

Реализация обмена данными

Подходы к интеграции и обмена информации между базами данных

Основные разделы:

* Настройка обмена данными для классификаторов между различными информационными базами
* Разработка планов обмена с отборами
* Интеграция прикладных решений через формат EnterpriseData

Разработка и использование библиотек

Основные подходы и методики к разработке переиспользуемой функциональности

Основные разделы:

* Разработка конфигураций с повторным использованием общего кода и объектов метаданных
* Имена объектов метаданных в иерархии библиотек
* Переопределяемые и поставляемые объекты библиотеки
* Отнесение библиотечных объектов к подсистемам
* Переопределение общих модулей в условиях иерархии библиотек
* Размещение сведений о настройках подсистемы
* Обеспечение совместимости библиотек
* Разработка ролей в библиотеках
* Обработчики обновления информационной базы (БСП)

Требования по локализации

Требования по разработке приложений используемых за пределами территории происхождения продукта и международных версий продукта

Основные разделы:

* Общие требования по локализации конфигурации
* Поставка международной версии конфигурации
* Интерфейсные тексты в коде: требования по локализации
* Запросы, динамические списки и отчеты на СКД: требования по локализации
* Форматирование даты, числа, Булево: требования по локализации
* Строковые константные выражения в коде: требования по локализации
* Элементы форм: требования по локализации
* Регламентные задания: требования по локализации
* Макеты: требования по локализации
* Денежные поля: требования по локализации
* Автогенерированные данные в информационной базе: требования по локализации.

Проектирование интерфейсов

Методики и рекомендации по построению интерфейса:

1. Интерфейс
   1. Обычные формы

[см. Приложение 4]

* 1. Управляемые формы
     1. 8.2

[см. Приложение 5]

* + 1. 8.3(Такси)

[см. Приложение 6]

Обработчики

* ПередЗаписью
* ПриКопировании
* ПроверкаЗаполнения
* ОбработкаЗаполнения
* Признак ОбменДанными.Загрузка

Асинхронное и Синхронное программирование. Многопоточность в 1с

Синхронные методы

Интерпретатор встроенного языка получает очередную команду для исполнения, приступает к ее выполнению. Этот процесс занимает какое-то время, после чего возвращается результат. Затем интерпретатор может переходить к следующей команде. И так до тех пор, пока все команды не будут выполнены.

Счетчик = 1;

ТекстСообщения = "Сообщение";

Сообщить(ТекстСообщения);

Таким образом, в каждый конкретный момент времени исполняется только одно действие, а несколько действий в рамках одного сеанса не могут выполняться одновременно.

Асинхронные методы

Асинхронные методы – это такие методы, выполнение которых не происходит синхронно с кодом, из которого они вызваны. В этом случае нет синхронизации по времени между вызывающим и вызываемым кодом. Вызывающий метод не дожидается завершения вызванного метода, а продолжает свое выполнение после вызова асинхронного метода. За счет этого не блокируется основной поток выполнения кода.

ПоказатьВопрос(Новый ОписаниеОповещения("ПоказатьВопросЗавершение", ЭтотОбъект), "Выполнить загрузку?", РежимДиалогаВопрос.ДаНет);

Сообщить("Начало загрузки");

[см. Приложение 7]

Описание оповещения

Используется для описания вызова процедуры программного модуля, который будет осуществлен при наступлении какого-либо ожидаемого события, такого как закрытие формы или немодального диалога

[см. Приложение 8]

Асинх Ждать Обещание

* Упрощение вызова
* Повешение читаемости кода

[см. Приложение 9]

Применение асинхронных методов

Работа с блокирующими окнами:

* Задать вопрос пользователю;
* Показать предупреждение;
* Открыть окно, блокирующее интерфейс, и обработать результат закрытия этого окна

Работа с файлами и каталогами:

* Получить файл с сервера;
* Получить каталог временных файлов;
* Скопировать файлы;
* Удалить файлы

Работа с механизмами криптографии:

* Получить перечень сертификатов;
* Проверить сертификат;
* Проверить действительность подписи;
* Зашифровать данные;
* Расшифровать зашифрованные данные

Работа с внешними компонентами:

* Подключить внешнюю компоненту;
* Обратиться к ее свойствам и методам.

Механизм заданий

Механизм заданий является одним из инструментов администрирования. Он включает в себя фоновые и регламентные задания. Фоновые задания позволяют инициализировать выполнения процедур общих модулей асинхронно (без ожидания завершения). Регламентные задания позволяют организовать автоматический вызов процедур общих модулей по расписанию.

Основное назначение этого механизма — выполнение административных действий по расписанию.

Возможности механизма заданий

* определение регламентных процедур на этапе конфигурирования системы;
* выполнение заданных действий по расписанию;
* выполнение вызова заданной процедуры или функции асинхронно, т. е. без ожидания ее завершения;
* мониторинг хода выполнения заданий;
* управление заданиями (отмена, блокировка выполнения и др.);
* возможность ожидания завершения одного или нескольких заданий.

Фоновые задания – это действия, производимые программой 1С незаметно для работающего в ней. Отслеживают их исполнение только по журналу регистрации. Особенностью такого задания является то, что при его выполнении не блокируется работа пользователя, он может продолжать вносить и редактировать данные, даже если процедура запущена.

Фоновые задания предназначены для выполнения прикладных задач асинхронно. Они могут порождать дочерние фоновые задания, например, для распараллеливания сложных вычислений по различным рабочим серверам кластера в клиент-серверном варианте работы.

Существует возможность ограничить выполнение фоновых заданий, имеющих одинаковые методы, по определенному прикладному признаку. Программное создание и управление фоновыми заданиями возможно из любого соединения пользователя с информационной базой системы 1С:Предприятие 8. Фоновое задание выполняется от имени пользователя, который его создал.

Регламентные задания — это общие объекты конфигурации. Они являются частью механизма заданий и позволяют автоматически выполнять процедуры на встроенном языке по расписанию.

# ЗАКЛЮЧЕНИЕ

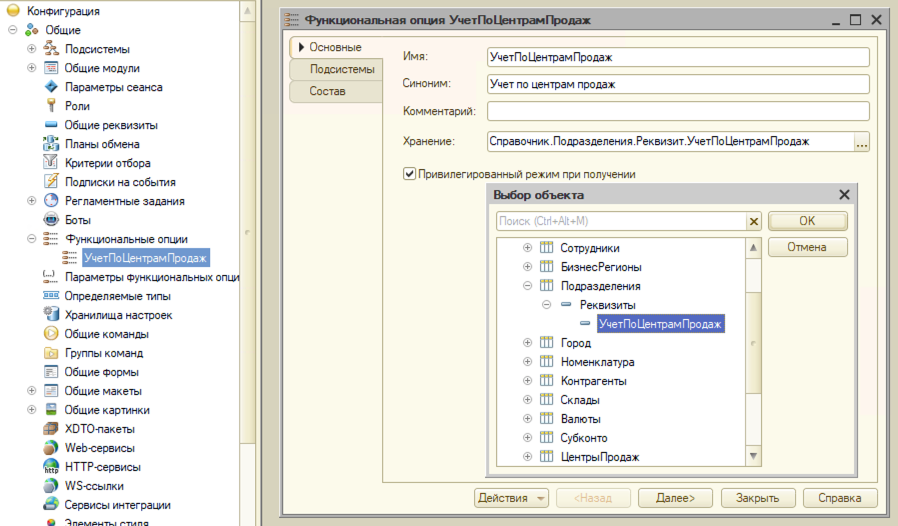
За период прохождения практики я получил умения и знания для использования 1С в профессиональной сфере деятельности. Ознакомился с основами разработки и проектирования на платформе 1С. Изучил архитектуру и функции 1С. Научился формировать запросы и отчёты. Получил опыт работы в профессиональном коллективе который смогу применить в будущей работе по специальности.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

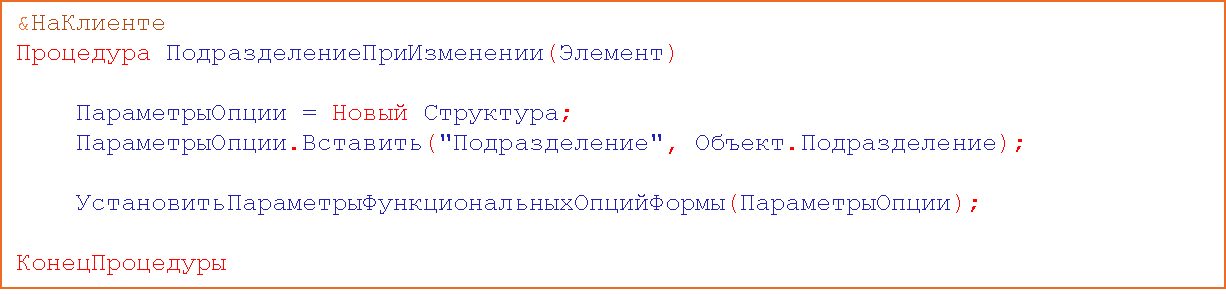
* ООО “Интеграл КТ” : сайт. – URL: <http://mbg-integral.ru/> (дата обращения: 17.06.2023)
* Методический материал в закрытом доступе (Курс обучения сотрудников “Профессиональная программа подготовки программистов 1С”) (дата обращения: 16.06.2023)
* Информационно-технологическое сопровождение (1С:ИТС): сайт. – URL: <https://its.1c.ru/db/v8std> (дата обращения: 18.06.2023)
* URL: <https://v8.1c.ru/platforma/obzor-arkhitektury-platformy> (дата обращения: 18.06.2023)
* URL: <https://center-comptech.ru/articles/sposoby-ucheta-nds-v-1s-buhgalteriya-8-3> (дата обращения: 18.06.2023)
* URL: <https://nizamov.studio/blog/page/7> (дата обращения: 25.06.2023)

# ПРИЛОЖЕНИЕ

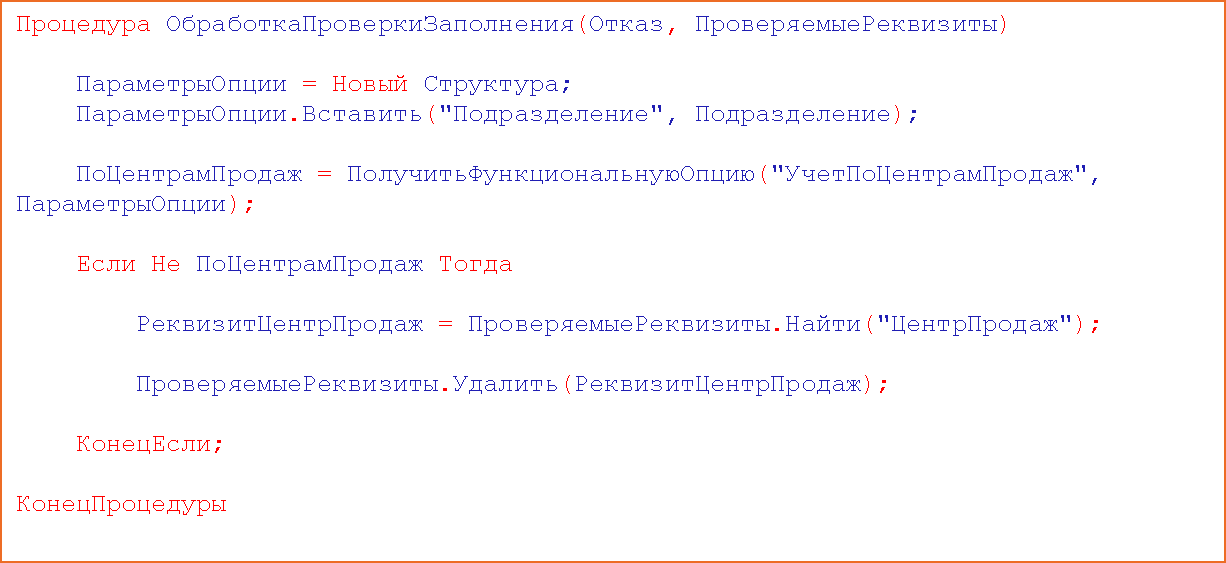
Приложение 1



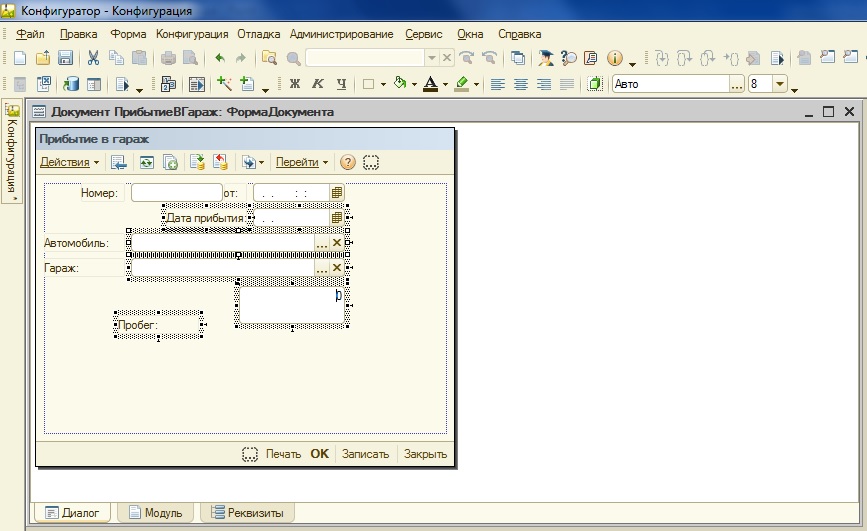
Приложение 2



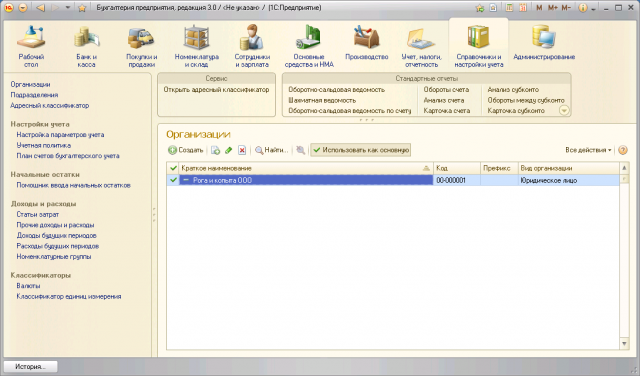
Приложение 3



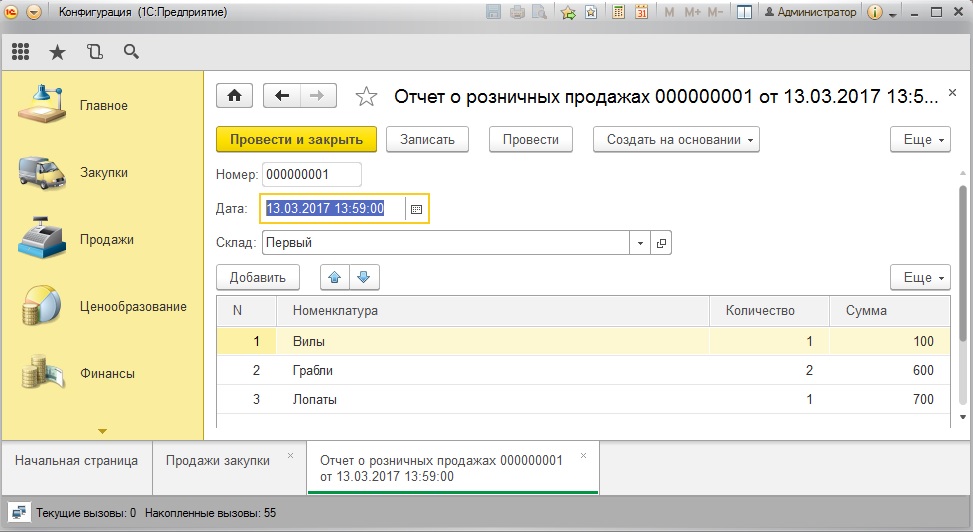
Приложение 4



Приложение 5



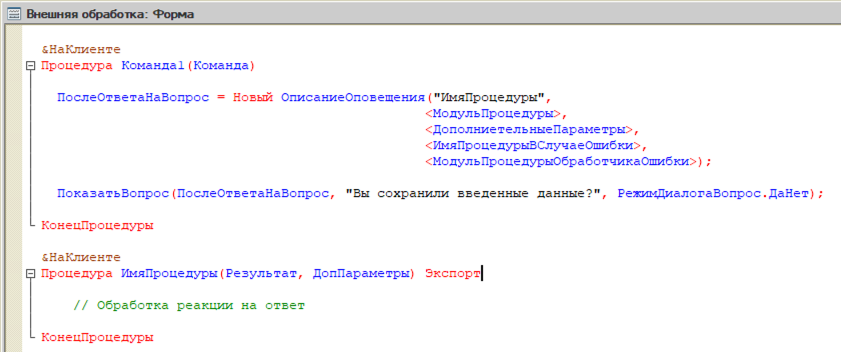
Приложение 6



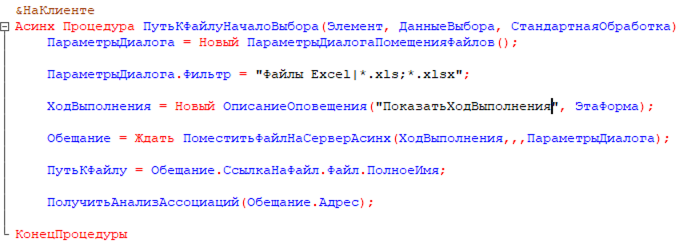
Приложение 7



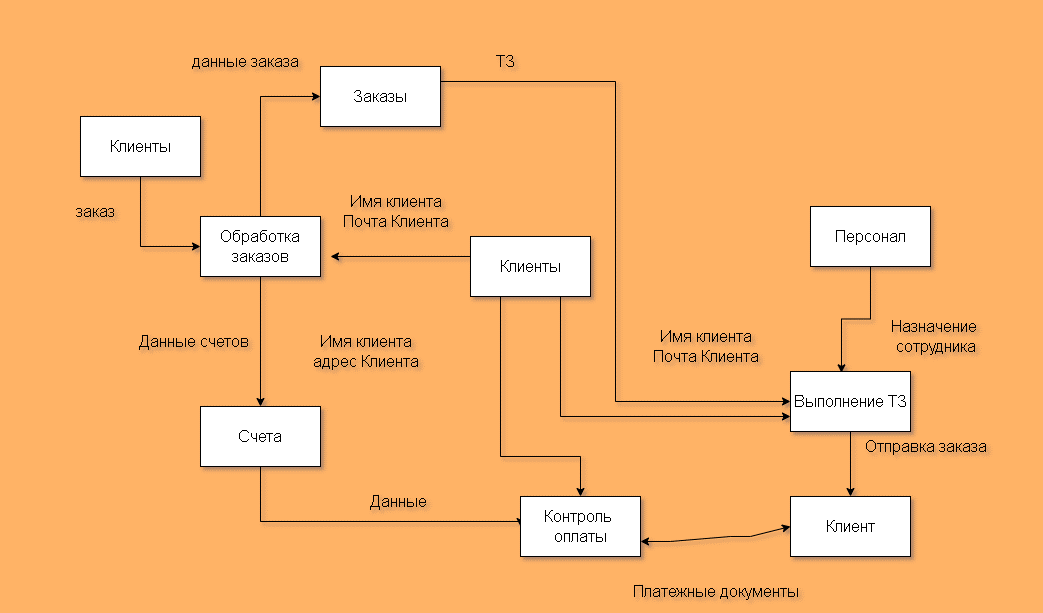
Приложение 8



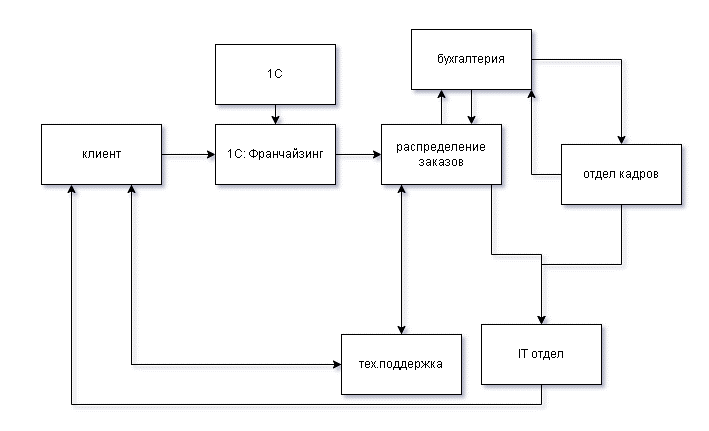
Приложение 9



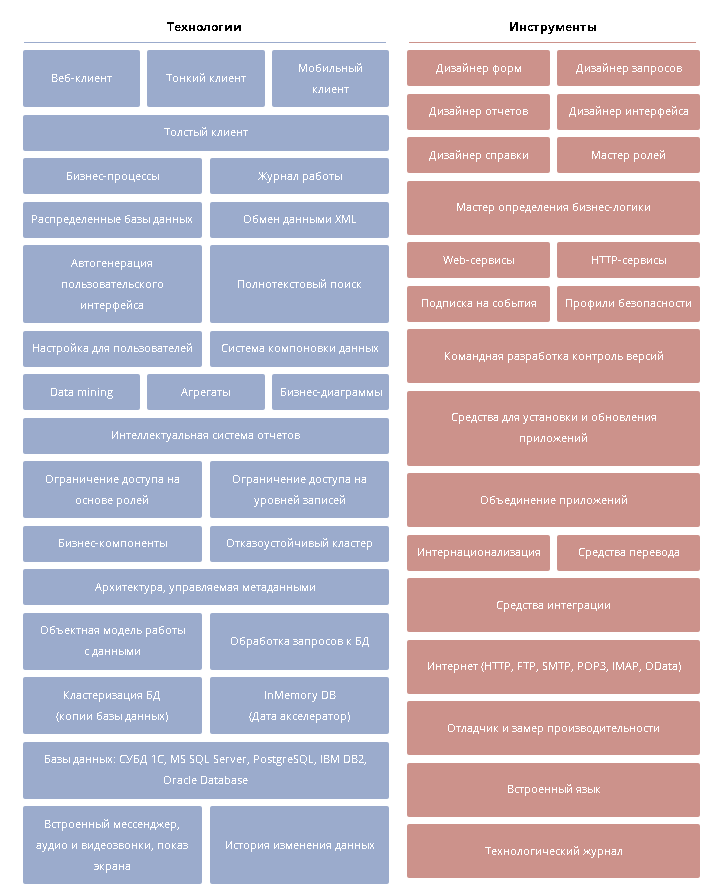
Приложение 10



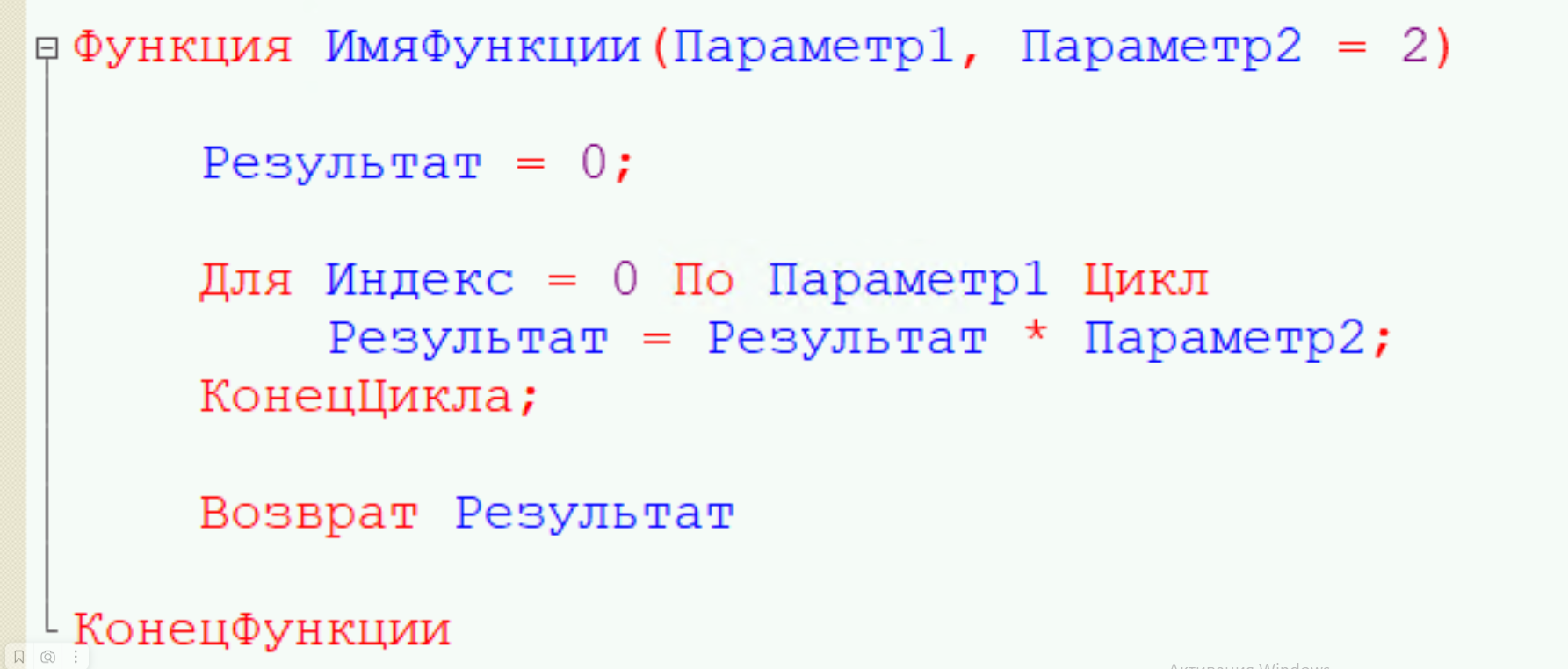
Приложение 11



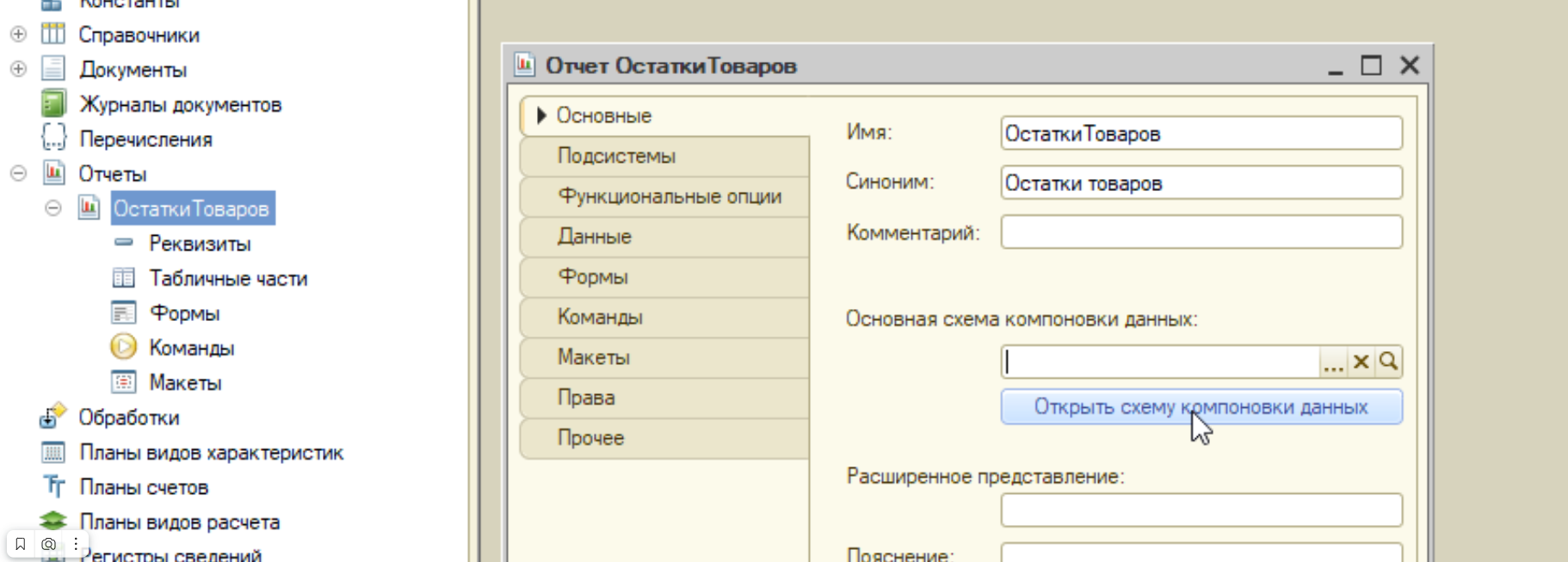
Приложение 12



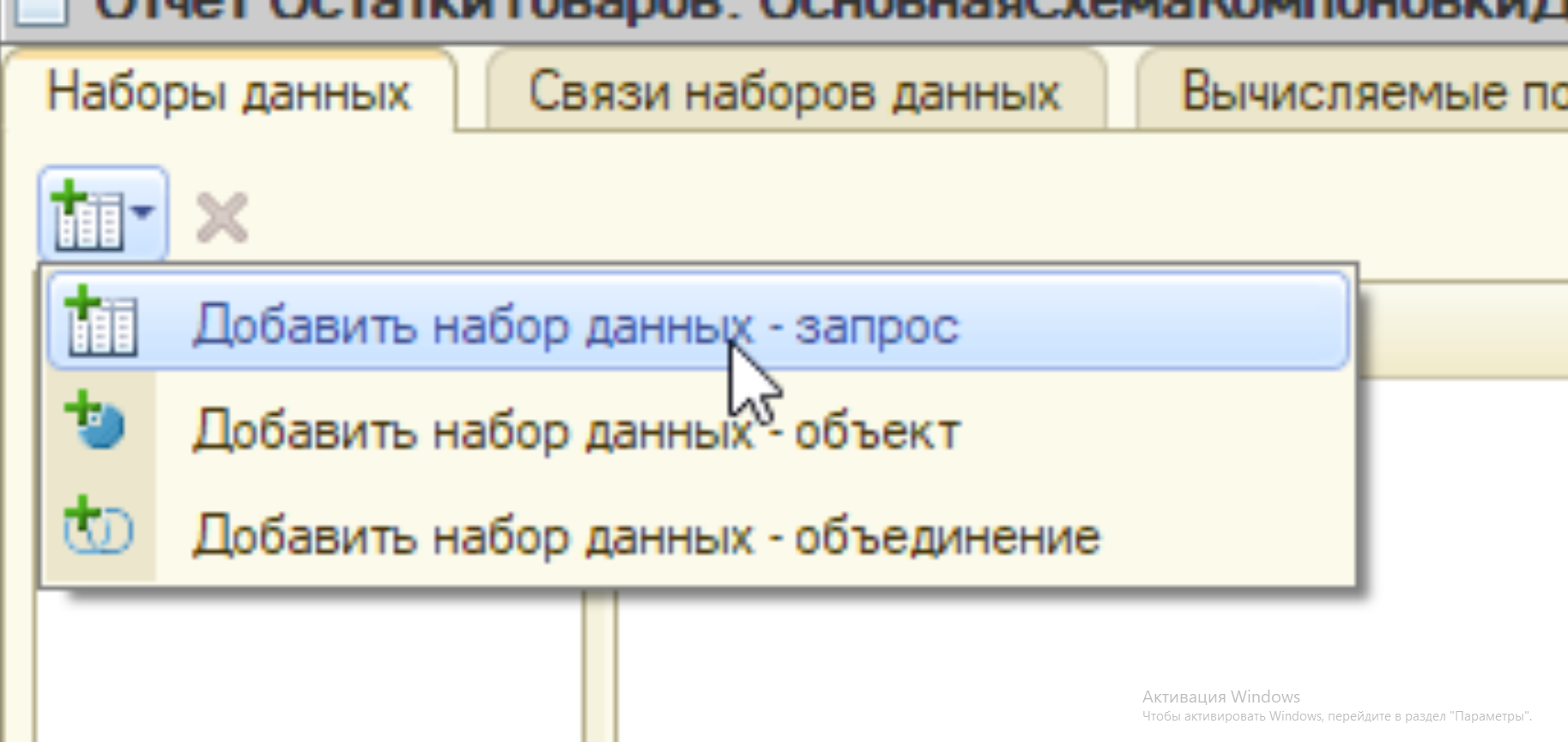
Приложение 13



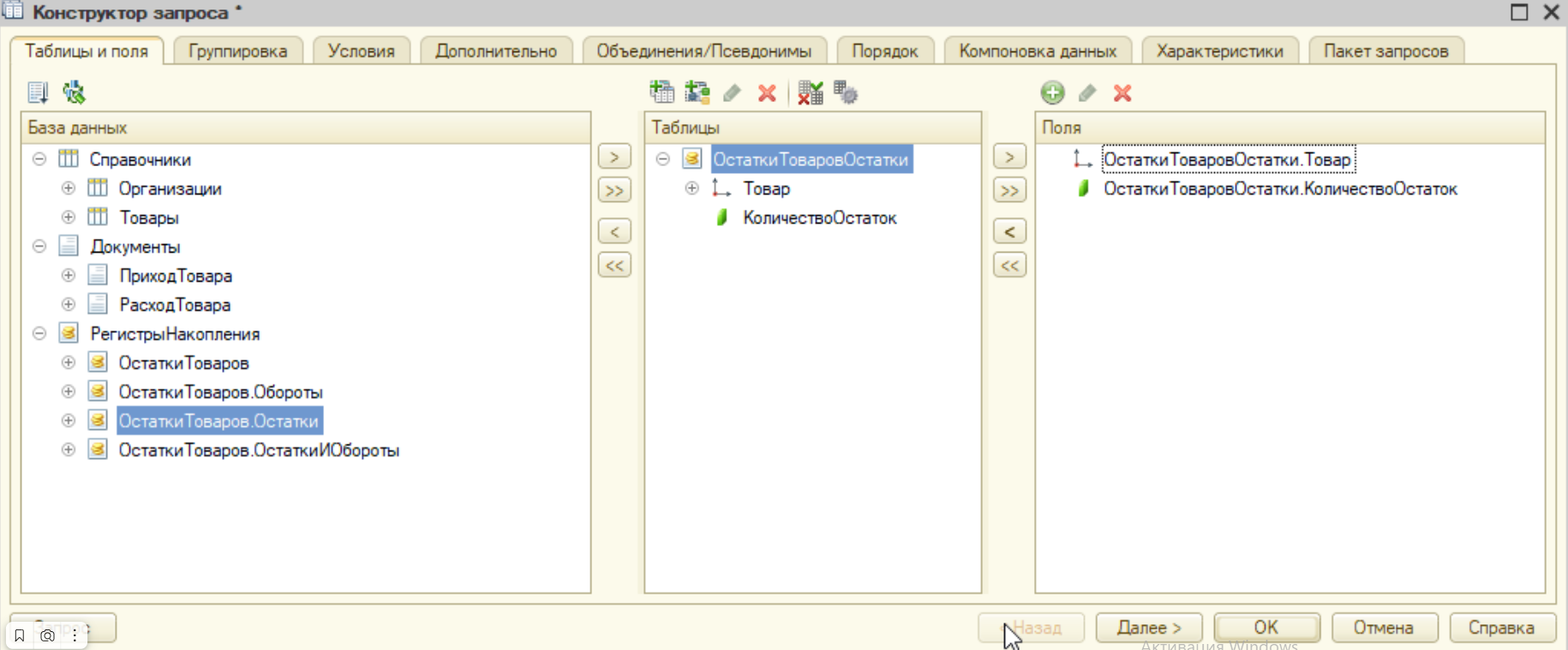
Приложение 14



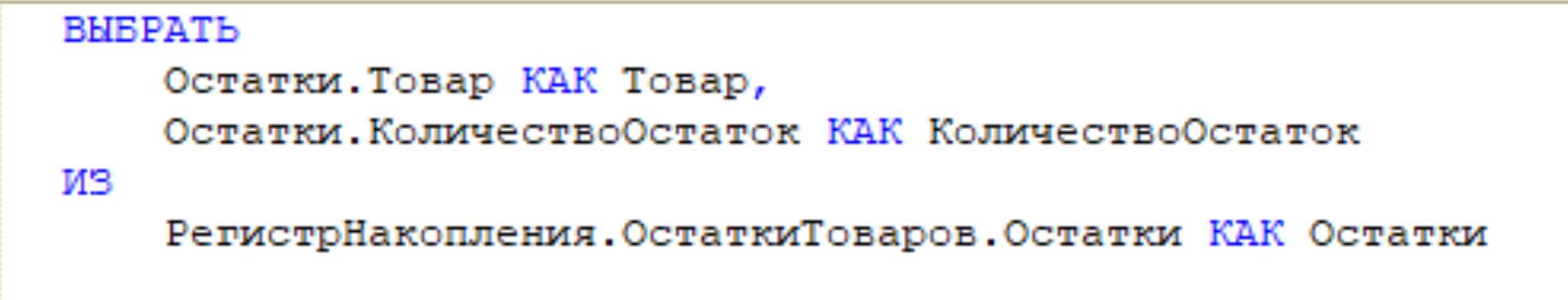
Приложение 15



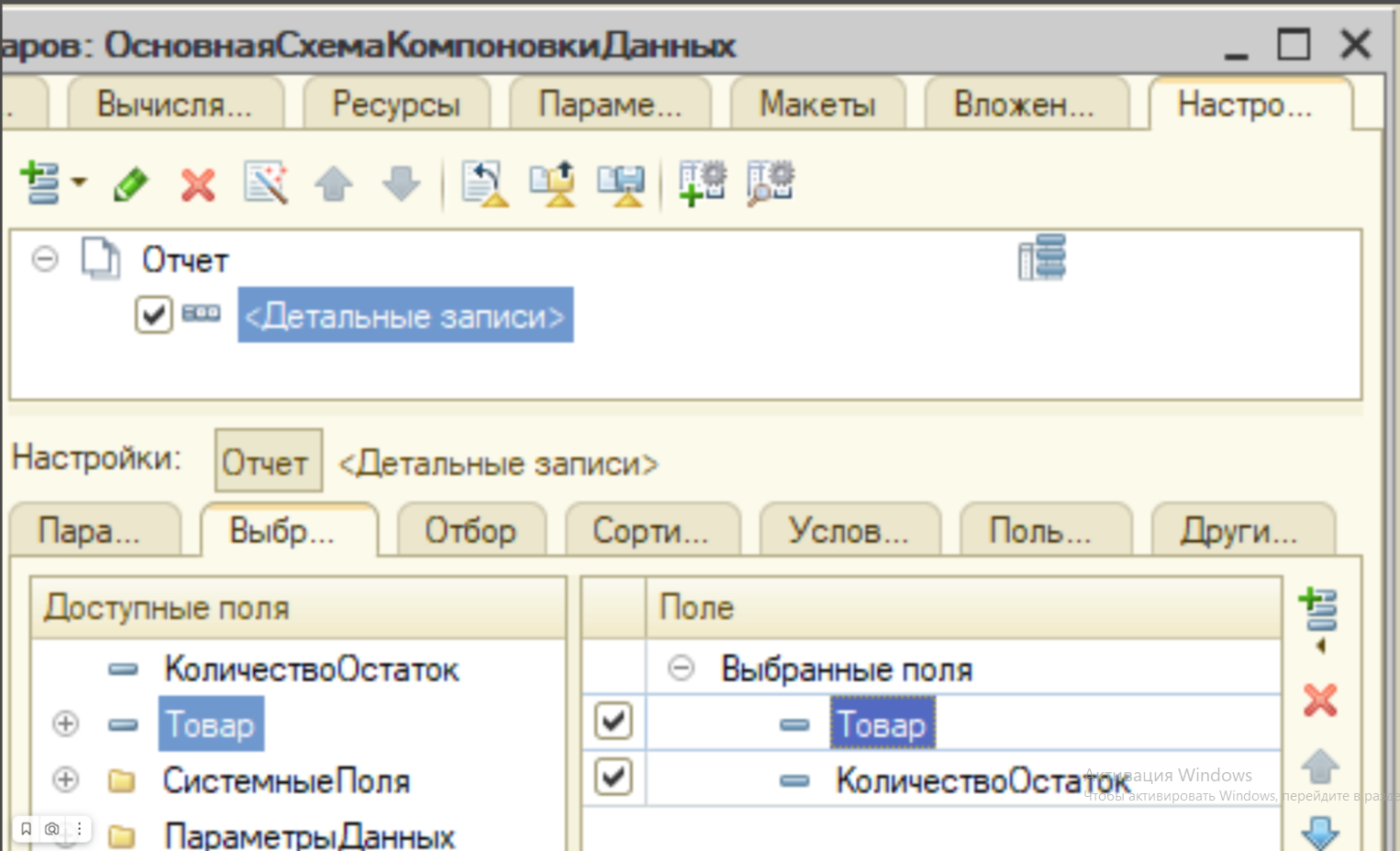
Приложение 16



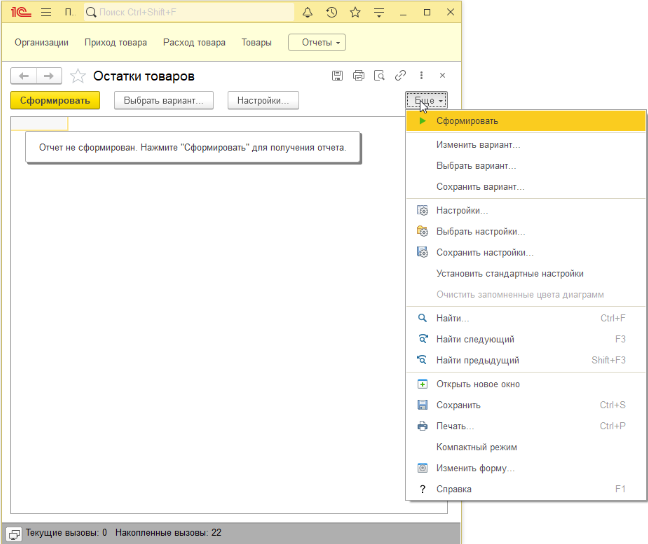
Приложение 17



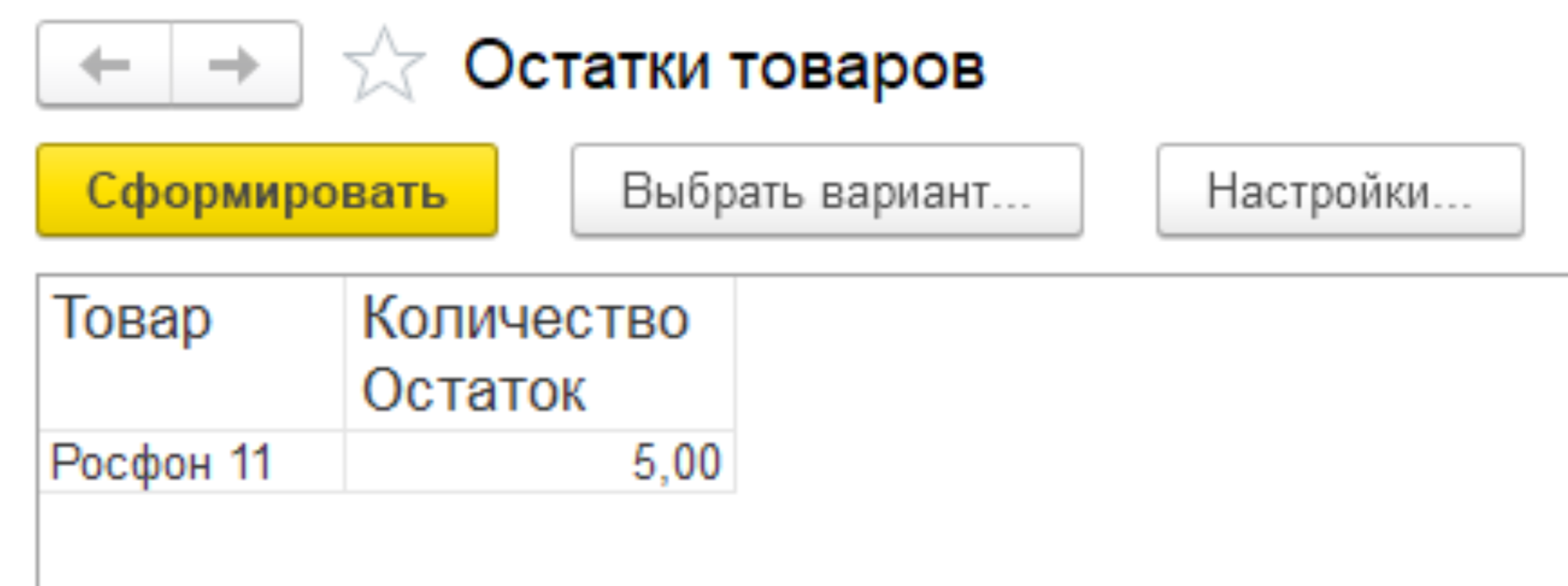
Приложение 18



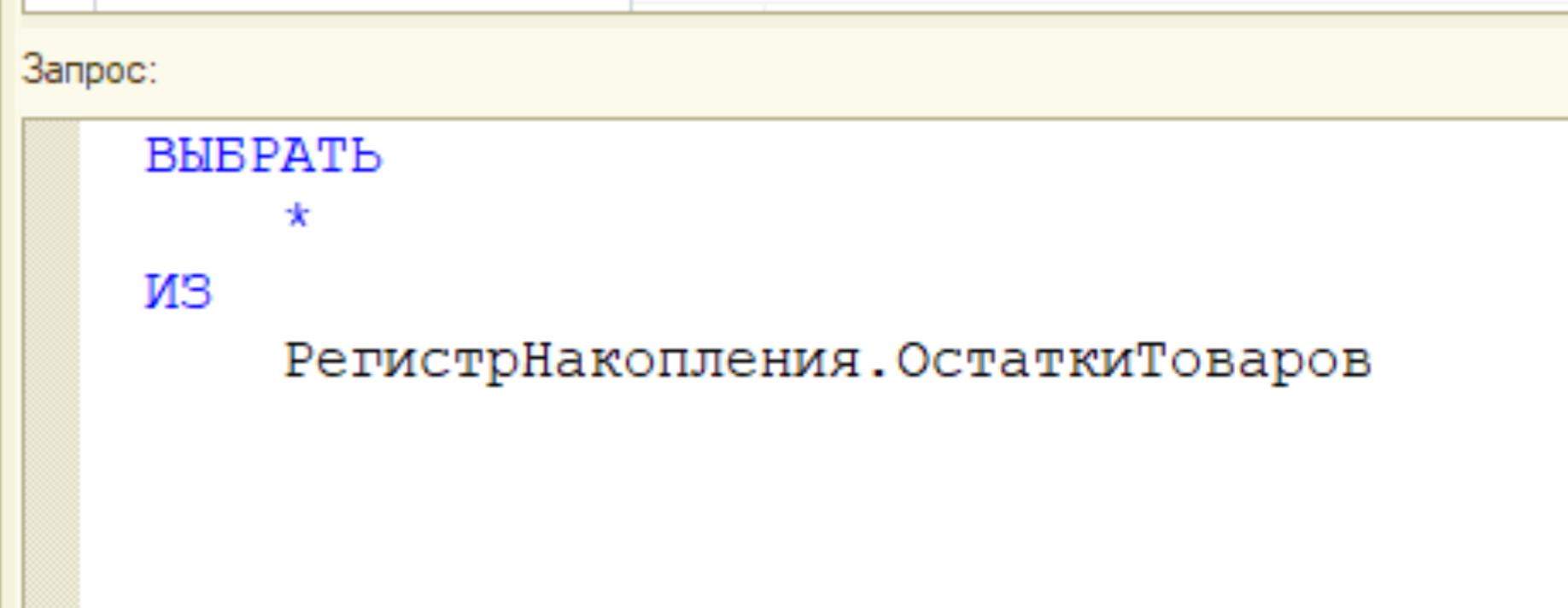
Приложение 19



Приложение 20



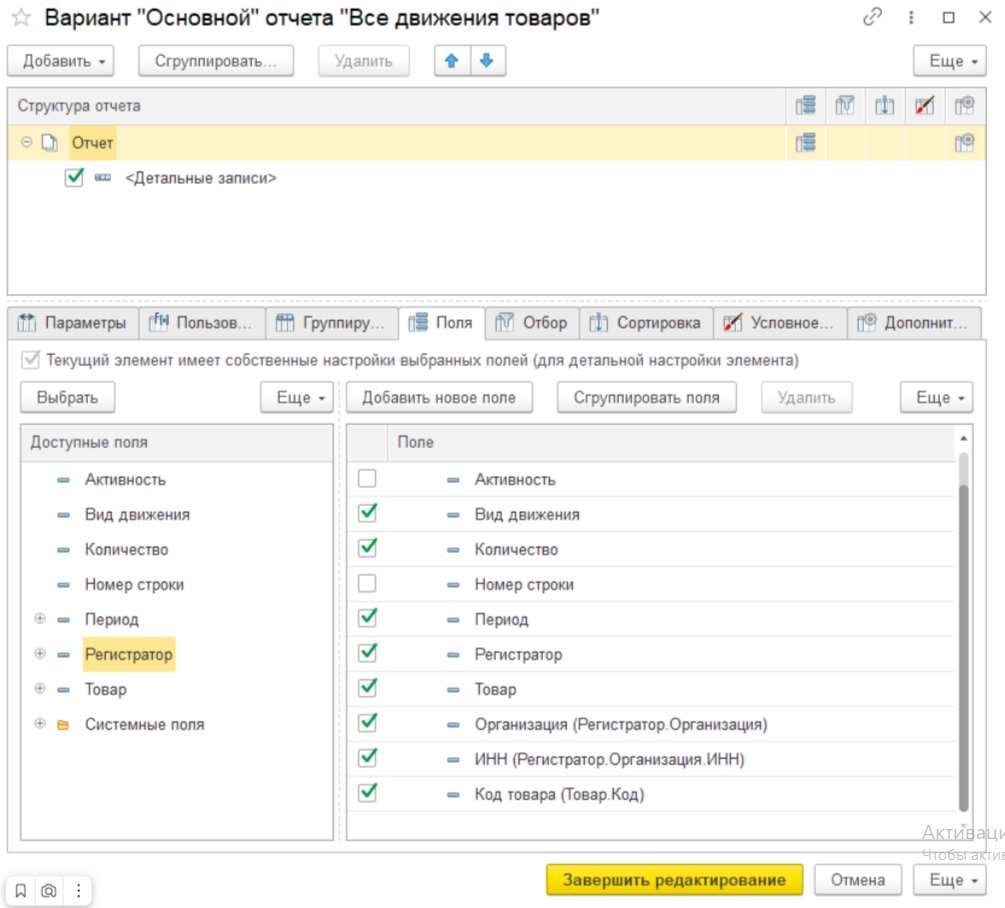
Приложение 21



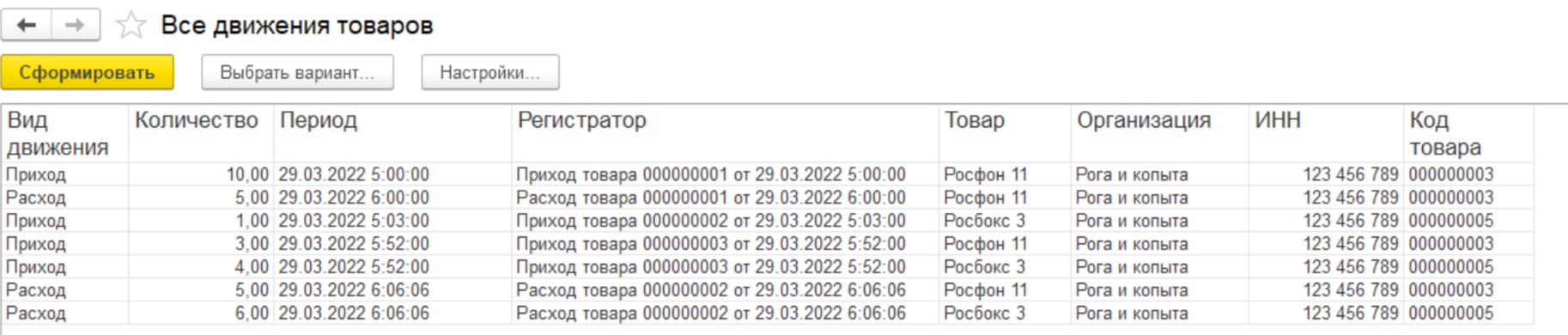
Приложение 22



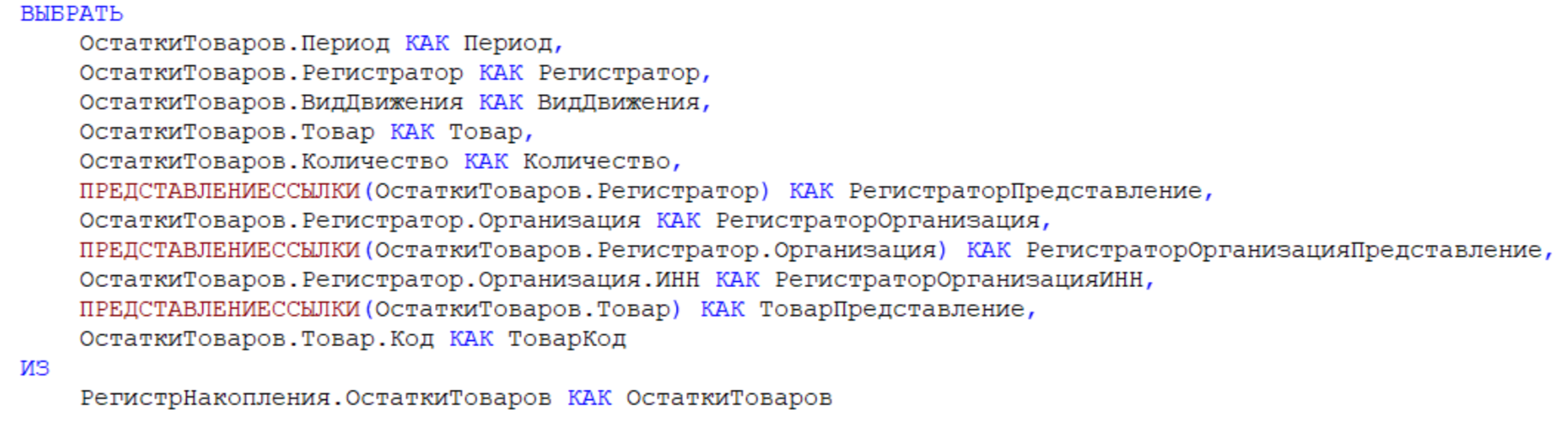
Приложение 23



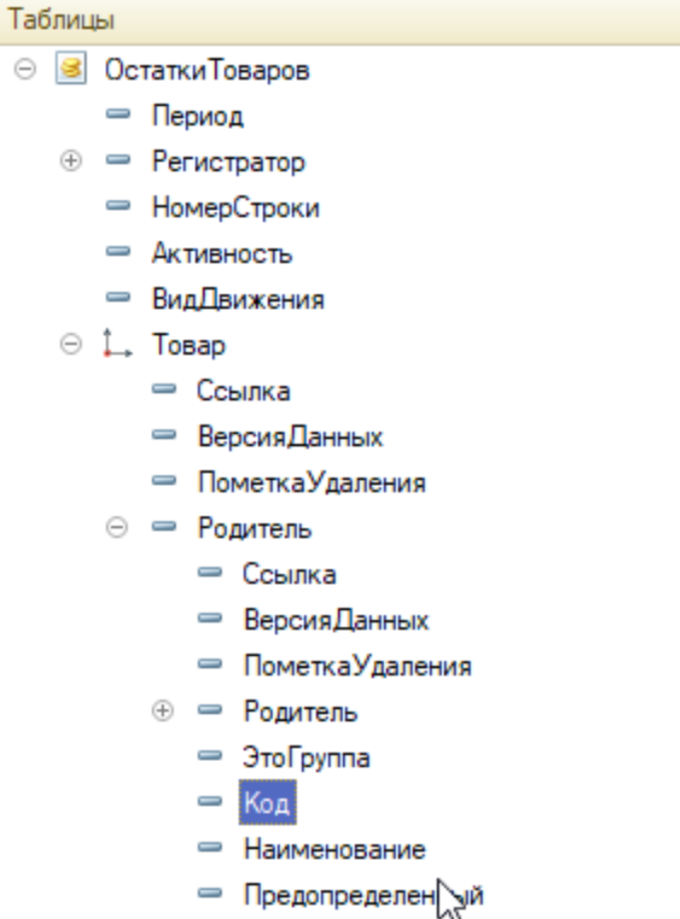
Приложение 24



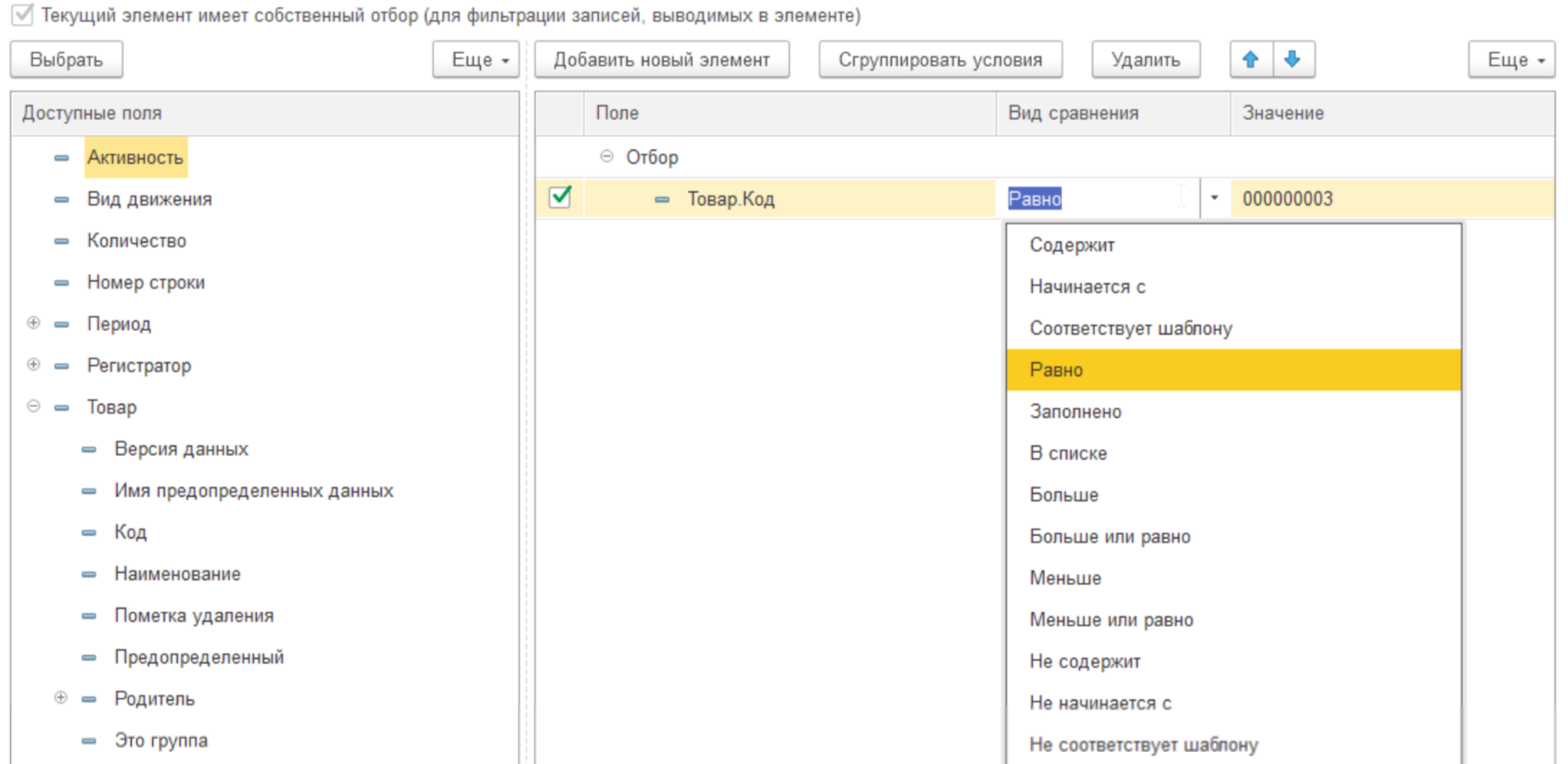
Приложение 25



Приложение 26



Приложение 27



Приложение 28

